Teleopit: A Full-Embodiment Humanoid Teleoperation System

Bingqian Wu[1,2], Zicheng Xu[1], Xianghui Fan[1], Dayu Li[1], Xiangru Huang[1]*

[1]Westlake University

[2]Shanghai Innovation Institute

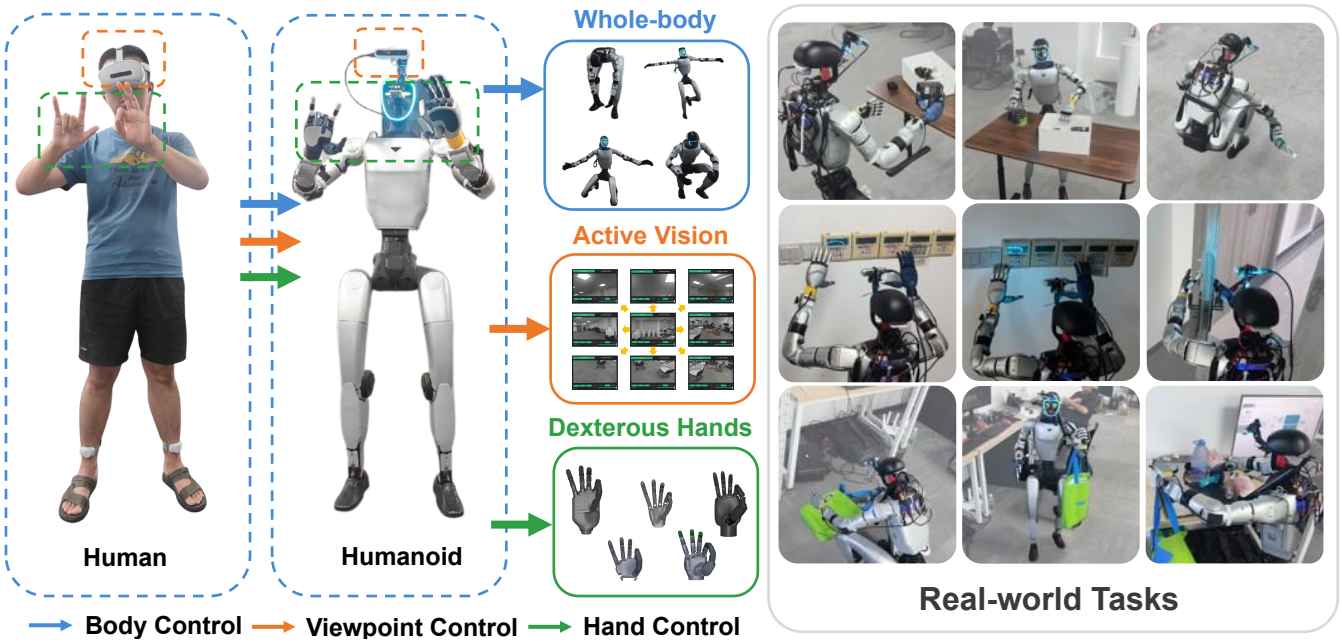

Figure 1: **Teleopit uses VR body, hand, and head tracking for full-embodiment humanoid teleoperation.** The interface coordinates dynamically feasible whole-body control, configurable dexterous hands, and active vision during real-world loco-manipulation.

Abstract

Humanoid teleoperation for demonstration collection requires coordinated whole-body motion, continuous dexterous hand control, and viewpoint control. Existing systems either simplify hand commands or depend on dedicated wearable sensors for fine-grained hand motion. We introduce Teleopit, a full-embodiment teleoperation system that maps body, hand, and head signals from VR to a humanoid body, configurable dexterous hands, and a 2-DoF active vision module. A history encoder and failure-aware rewind sampling improve the motion tracker on both motion-capture and live VR references. An optimization-based hand retargeter combines normalized finger directions, fingertip closure, and thumb-frame alignment to map human hand motion to different dexterous hands without tuning hand-specific objective or solver hyperparameters. Component experiments evaluate tracking success rate and retargeting behavior, while real-robot teleoperation demonstrates coordinated locomotion, manipulation, and viewpoint control. ACT and GR00T N1.7 policies trained on 96 successful demonstrations collected with Teleopit achieve task success rates of 90.0% and 95.0%, respectively, when deployed on the humanoid. The project page is available at https://botrunner64.github.io/teleopit-page.

1 Introduction

Humanoid teleoperation provides a direct interface for collecting real-world demonstrations for data-driven robot learning. Vision-language-action models and robot foundation models increasingly benefit from broad real-world robot datasets [1–7]. Unlike tabletop manipulation, humanoid demonstrations require balance, locomotion, whole-body manipulation, continuous hand motion, and viewpoint control to remain coordinated throughout a task.

Existing interfaces address only parts of this coordination problem. VR systems can provide body, hand, and head signals without dedicated motion capture equipment, but current humanoid systems commonly use only a subset of these signals: some control the upper body and hands without locomotion, while others control the whole body but reduce the hands to discrete gripper commands [8, 9]. Systems that achieve contin-

*Corresponding author.

uous whole-body and dexterous hand control instead rely on custom inertial suits and gloves [10]. General XR frameworks simplify device integration but do not by themselves provide a dynamically controlled humanoid pipeline [11]. Moreover, dexterous hand retargeting must accommodate different link lengths, joint layouts, and thumb structures; existing optimization methods require hand-dependent geometric scaling, while learned mappings require training data for the target hand [12–14]. These limitations leave no single workflow that combines VR sensing, dynamically feasible whole-body control, continuous cross-hand retargeting, viewpoint control, and recording.

We introduce Teleopit (Figure 1), a full-embodiment humanoid teleoperation system that uses VR as a unified source of operator intent. A body tracker maps the operator's motion to dynamically feasible humanoid control, an optimization-based retargeter maps hand tracking to configurable dexterous hands, and an active vision module maps head motion to viewpoint control. An asynchronous runtime connects these components with visual feedback and recording, enabling **joint whole-body, hand, and viewpoint control** while keeping each control stream responsive at its respective rate.

The whole-body tracker targets stable control from temporally correlated and occasionally noisy VR references. Its **history encoder** integrates recent reference and proprioceptive observations so that the policy can infer motion context that is unavailable from a single frame. During training in mjlab [15], **failure-aware rewind sampling** returns failed rollouts to the difficult segment that preceded termination, increasing exposure to transitions that interrupt teleoperation.

The optimization-based hand retargeter addresses morphology differences, building on vector-based dexterous teleoperation [12, 13]. Normalized finger directions remove bone-length scale from the primary objective, while fingertip-distance and thumb-frame objectives encode pinch closure and thumb opposition. After semantic links are specified for a robot hand, the same objective weights, activation thresholds, and solver settings map human motion to different hand morphologies, providing **cross-embodiment transfer without hand-specific tuning**.

We evaluate the motion tracker on held-out motion-capture and live VR references, including comparisons with recent trackers and ablations of the history encoder and rewind sampling. Hand experiments examine cross-embodiment pose transfer and the effects of the distance and frame objectives. We further conduct real-robot teleoperation on tasks that coordinate whole-body motion, continuous hand control, and viewpoint control. Finally, we use 96 successful demonstrations collected with Teleopit to train ACT and GR00T N1.7. When deployed on the humanoid, the resulting policies achieve task success rates of 90.0% and 95.0%, respectively, on a bottle-placement task.

Our contributions are summarized as follows:

- A VR-driven humanoid teleoperation system for collecting demonstrations with **joint whole-body, hand, and viewpoint control**.
- An optimization-based hand retargeter providing **cross-embodiment transfer without hand-specific tuning**.
- A motion tracker with a **history encoder** and **failure-aware rewind sampling** for robust live VR tracking.

2 Related Work

2.1 Humanoid Teleoperation and Data Collection

Humanoid teleoperation has progressed from kinematic motion transfer and balance-aware inverse kinematics to learned controllers that track human motion in real time [16–21]. Recent systems extend control beyond the body by combining VR, visual feedback, or dexterous hands for loco-manipulation [8, 22–24]. Robot-free interfaces offer higher collection throughput but translate human demonstrations to the robot offline, while general XR frameworks provide device and robot interfaces without a complete dynamically controlled humanoid workflow [11, 25, 26].

TWIST2 and HumDex are the closest complete systems. TWIST2 integrates PICO body sensing, full-body tracking, and active vision, but controls its robot hands as discrete grippers [9]. HumDex continuously controls a dexterous hand and the humanoid body using a custom inertial suit and commercial gloves [10]. Teleopit instead obtains body, hand, and head signals from VR and integrates whole-body tracking, continuous dexterous hand control, and viewpoint control.

2.2 Humanoid Motion Tracking

Physics-based motion tracking uses reference-conditioned policies, motion priors, and unified representations to reproduce diverse humanoid motions [27–33]. Recent trackers improve coverage through adaptive sampling, mixtures of experts, temporal context, specialized policies, or larger training corpora [9, 34–39]. Teleopit combines temporal history with failure-aware sampling to improve the success rate of a single tracker on both motion-capture and live VR references.

2.3 Dexterous Hand Retargeting

Dexterous hand retargeting must map human motion across differences in link length, joint layout, and actuation. Geometric methods optimize fingertip positions, joint and Cartesian objectives, or selected hand vectors, but typically depend on calibration or morphology-specific scaling [12, 13, 40–43]. Learned methods shift computation from online optimization to target-hand training, paired samples, or guided data collection [10, 14, 44, 45].

Task-oriented approaches additionally preserve contact, grasp function, or dynamic feasibility for specific manipulation settings [46–49]. Teleopit instead uses a shared online optimization objective: normalized finger directions handle morphology differences, while fingertip distance and thumb frame objectives preserve pinch closure and thumb opposition.

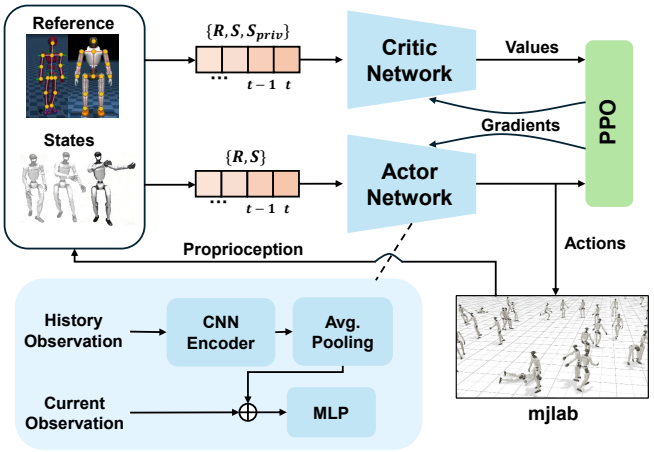

Figure 2: **Whole-body motion tracking pipeline.** The actor combines current proprioception and temporal history, while the training-only critic also uses privileged state. PPO optimizes predicted joint targets.

New hands require semantic link mapping but no hand-specific objective or solver hyperparameter tuning.

3 Method

3.1 Overview

Teleopit maps body, hand, and head signals from PICO to a humanoid robot in real time. The input comprises a 24-joint body skeleton $\mathbf{s}^{\text{body}}$, 26 keypoints for each hand $\mathbf{s}^{\text{hand}}$, and a head pose $\mathbf{s}^{\text{head}}$. The outputs are body joint targets $\mathbf{a}^{\text{body}}$, dexterous hand commands $\mathbf{a}^{\text{hand}}$, and a 2-DoF viewpoint command $\mathbf{a}^{\text{cam}}$.

The system contains a learned whole-body motion tracker, an optimization-based hand retargeter, and a runtime that integrates sensing, control, visual feedback, and recording. The following sections describe these components and the motivation for their designs.

3.2 Whole-Body Motion Tracking

Architecture. The tracker must convert a human reference into dynamically feasible robot motion while remaining deployable with onboard state estimation. As shown in Figure 2, we train a single policy with PPO [50] in the GPU-accelerated MuJoCo simulator mjlab [15]. The actor receives robot proprioception and a reference frame and predicts a 29-dimensional action. We convert the action to joint targets as

$$\mathbf{q}^{\text{target}} = \text{clip}(\mathbf{a}, -c, c) \odot \mathbf{s} + \mathbf{q}^{\text{def}}, \tag{1}$$

where $\mathbf{s}$ is a per-joint action scale and $\mathbf{q}^{\text{def}}$ is the default pose. A PD controller tracks these targets at 200 Hz while the policy runs at 50 Hz. Tracking is anchored at the torso and covers 14 body links.

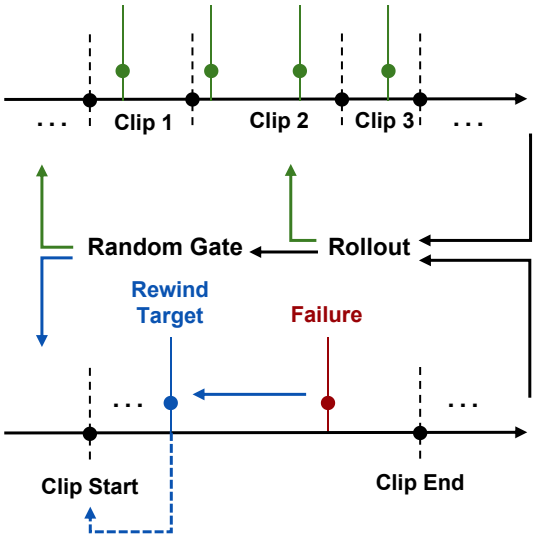

Figure 3: **Failure-aware rewind sampling.** Failed rollouts retain the clip and rewind before the failure, focusing subsequent updates on difficult transitions.

History Encoder. A single observation frame does not reveal recent contact changes or body momentum, which are important when VR references vary over time. The History Encoder stacks the actor observation over the previous H=10 control steps and applies a one-dimensional temporal convolution followed by global average pooling. The resulting latent is concatenated with the current observation before the policy MLP, as illustrated in Figure 2. This supplies motion context without an additional estimator or privileged teacher. The critic receives additional simulation state during training, but these signals are not used by the actor.

Failure-Aware Rewind Sampling. Uniform sampling spends most updates on common motion while rarely revisiting the difficult transitions that terminate teleoperation. Motivated by adaptive sampling of difficult motion [34], we use the failure signal itself to identify hard segments. Training initially samples clips in proportion to duration and resets the robot to the sampled reference state. When a rollout terminates early, the strategy in Figure 3 retains the clip with high probability and rewinds the reference time by a random offset. The policy therefore practices the transition immediately before failure more frequently, without a separate difficulty model or staged curriculum.

Observations and Rewards. The tracker combines deployable actor observations with privileged critic signals and a dense whole-body tracking objective.

Observation Design The observation design separates deployable robot state from training-only information. Table 1 lists the per-step inputs. The actor receives the reference motion, joint state, base angular velocity, projected gravity, and previous action. These signals describe the target and

Table 1: Tracker observations. Noise is the half-width of uniform training corruption. A and C denote actor and critic inputs; the lower block is critic-only.

Term	Dim	Noise	A / C
Reference joint position	29	–	A, C
Reference joint velocity	29	–	A, C
Reference anchor orientation (6D)	6	0.05	A, C
Reference anchor lin. velocity	3	–	A, C
Reference anchor ang. velocity	3	–	A, C
Reference projected gravity	3	–	A, C
Reference anchor height	1	–	A, C
Joint position (rel. to default)	29	0.01	A, C
Joint velocity	29	0.5	A, C
Base angular velocity	3	0.2	A, C
Projected gravity	3	0.05	A, C
Previous action	29	–	A, C
Reference anchor position	3	–	C
Base linear velocity	3	–	C
Tracked-body position (14)	42	–	C
Tracked-body orientation (14)	84	–	C

measured response without requiring global body poses. A ten-frame history provides recent motion context.

The critic additionally receives global reference position, base linear velocity, and the poses of the 14 tracked links, but the deployed actor never sees them. Uniform corruption is applied to the channels marked in Table 1; reference joint targets and the previous action remain clean. This split keeps the actor inputs deployable while enabling privileged value estimation during training.

Reward Design The reward combines global tracking, articulated pose tracking, and deployment-oriented regularization. In Table 2, anchor terms preserve the torso trajectory, body and joint terms constrain the full pose and dynamics, and the wrist-palm point directly targets hand placement for downstream dexterous control.

Gaussian scales determine the informative error range of each tracking term. The remaining terms discourage rapid commands, joint-limit violations, self-collision, and high ankle acceleration, while rewarding survival. Together, the terms encode both pose tracking and motion regularity rather than a single joint-space error.

Domain Randomization Design The domain randomization mechanisms cover persistent physical variation, sensing noise, and transient disturbances. At each environment reset, we randomize friction, torso center of mass, link inertia, and joint offsets. Per-step observation noise perturbs selected measurements, while periodic base-velocity perturbations expose the policy to transient disturbances. Table 3 lists the range of each perturbation.

Reference Sampling and Termination Reference-state initialization resets each environment to a sampled pose with the

Table 2: Tracker reward. Tracking terms use a Gaussian kernel over squared error e; regularization terms use the listed penalties.

Term	Expression	Weight
Tracking $(r = \exp(-e/\sigma^2))$		
Anchor position	$\sigma = 0.3$	0.5
Anchor orientation	$\sigma = 0.4$	0.5
Anchor linear vel.	$\sigma = 1.0$	1.0
Anchor angular vel.	$\sigma = 3.0$	1.0
Body position	$\sigma = 0.3$	1.0
Body orientation	$\sigma = 0.4$	1.0
Body linear vel.	$\sigma = 1.0$	1.0
Body angular vel.	$\sigma = 3.14$	1.0
Joint position	$\sigma = 0.5$	1.0
Joint velocity	$\sigma = 3.0$	0.5
Wrist palm point	$\sigma = 0.12$	1.0
Regularization		
Survival	$\mathbf{1}$ (alive)	3.0
Action rate	$\|a_t - a_{t-1}\|^2$	−0.5
Joint-limit violation	$\sum_j \max(0, q_j - q_j^{\text{lim}})$	−10.0
Self-collision	$\mathbf{1}[f_{\text{self}} > 1\text{N}]$	−0.1
Ankle joint accel.	$\|\ddot{q}_{\text{ankle}}\|^2$	-2.5×10^{-6}

perturbations in Table 3, and clips are sampled in proportion to their valid duration. When a rollout terminates early, rewind sampling retains the same clip with probability 0.8 and moves the reference to an earlier state near the failed transition.

An episode terminates when the anchor height error exceeds 0.25 m, the anchor orientation error exceeds 1.0 rad, or the height error of an ankle or wrist exceeds 0.25 m. These criteria supply the failure event used by rewind and prevent severely diverged states from dominating subsequent updates. Thus, domain randomization broadens the physical and sensory conditions encountered during training, while reference-state initialization and rewind determine where training effort is spent along each motion clip.

3.3 Dexterous Hand Retargeting

Formulation and Notation. Human and robot hands differ in bone lengths, joint layouts, and thumb articulation. We therefore optimize geometric relations that can be defined on both embodiments, as shown in Figure 4. Let $\mathbf{q}$ denote the robot hand configuration, $\mathbf{v}_i(\mathbf{q})$ a robot finger segment, and $\hat{\mathbf{d}}_i$ the corresponding unit human segment direction. For fingertip pair k, $\rho_k(\mathbf{q})$ and $\hat{\rho}_k$ denote robot and target distances. Robot thumb-frame axes are $\mathbf{e}_p(\mathbf{q}), \mathbf{e}_s(\mathbf{q})$, with human references $\hat{\mathbf{e}}_p, \hat{\mathbf{e}}_s$. The retargeted pose is

$$\mathbf{q}^\star = \arg\min_{\mathbf{q}} \mathcal{L}_{\text{dir}} + \mathcal{L}_{\text{dist}} + \mathcal{L}_{\text{frame}} + \lambda \|\bar{\mathbf{q}} - \bar{\mathbf{q}}_{\text{prev}}\|^2, \quad (2)$$

subject to the robot joint limits. The final term smooths the expanded joint configuration $\bar{\mathbf{q}}$ over time.

Table 3: Domain randomization, observation noise, and reference-state initialization for the tracker. $\pm x$ denotes a uniform draw over $[-x, x]$.

Parameter	Range
Dynamics (startup)	
Ground friction	$[0.3, 1.6]$
Torso CoM offset (x, y, z) [m]	$(\pm 0.025, \pm 0.05, \pm 0.05)$
Link pseudo-inertia scale α	$[-0.1, 0.45]$
Default joint-position offset [rad]	± 0.01
Observation noise (uniform)	
Joint position [rad]	± 0.01
Joint velocity [rad/s]	± 0.5
Base angular velocity [rad/s]	± 0.2
Projected gravity	± 0.05
Anchor orientation	± 0.05
Random push (every 4–6 s)	
Base linear velocity (x, y, z) [m/s]	$(\pm 0.5, \pm 0.5, \pm 0.2)$
Base angular velocity (r,p,y) [rad/s]	$(\pm 0.52, \pm 0.52, \pm 0.78)$
Reference-state init.	
Root position (x, y, z) [m]	$(\pm 0.05, \pm 0.05, \pm 0.01)$
Root orientation (r,p,y) [rad]	$(\pm 0.1, \pm 0.1, \pm 0.2)$
Root velocity	same as push
Joint position [rad]	± 0.1

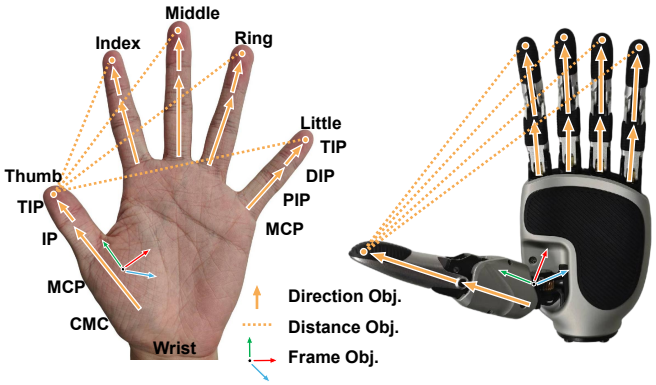

Figure 4: **Hand retargeting objectives.** Direction, fingertip-distance, and thumb-frame objectives map the human pose to a robot hand with different morphology.

Direction Objective. The orange arrows in Figure 4 align corresponding segments through

$$\mathcal{L}_{\text{dir}} = \sum_i w_i \left(1 - \frac{\mathbf{v}_i(\mathbf{q})}{\|\mathbf{v}_i(\mathbf{q})\|} \cdot \hat{\mathbf{d}}_i\right). \tag{3}$$

Normalizing both segments removes bone length from the comparison. Unlike scaled keyvector objectives [12, 13, 51], this objective does not require a human-to-robot scale for each hand.

Distance Objective. Direction alignment does not guarantee fingertip closure. The dotted lines in Figure 4 identify thumb-to-fingertip pairs, for which we use

$$\mathcal{L}_{\text{dist}} = \sum_k w_k a_k \max(0, \rho_k(\mathbf{q}) - \hat{\rho}_k)^2. \tag{4}$$

The activation a_k increases smoothly when the human fingertips approach a 4 cm contact threshold. The one-sided objective closes a commanded pinch without pushing fingertips apart during free hand motion.

Frame Objective. A single thumb segment does not determine opposition about its axis. The axes at the thumb base in Figure 4 define a local frame from the thumb CMC, thumb MCP, and index MCP. Gram–Schmidt orthonormalization produces the two reference axes, which are aligned by

$$\mathcal{L}_{\text{frame}} = w_p \left(1 - \mathbf{e}_p(\mathbf{q}) \cdot \hat{\mathbf{e}}_p\right) + w_s \left(1 - \mathbf{e}_s(\mathbf{q}) \cdot \hat{\mathbf{e}}_s\right). \tag{5}$$

This objective resolves thumb roll and supports both precision and power grasp poses.

Optimization and Hand Configuration. The 21 human keypoints are expressed in a wrist frame derived from the wrist, index MCP, and middle MCP. We solve Equation (2) for each frame with SLSQP, analytic kinematic gradients, and the previous solution as the initial state. The objective terms reference semantic link names, so a new hand only specifies the corresponding robot links. For mechanically coupled hands, independent coordinates $\mathbf{q}$ are expanded through a coupling map $\bar{\mathbf{q}} = \phi(\mathbf{q})$, and gradients propagate through ϕ. All configured hands share the numerical objective weights, activation thresholds, and solver settings; Section 4.2 evaluates this cross-embodiment transfer.

3.4 System Integration

Figure 5 summarizes the runtime that connects the motion tracker, hand retargeter, and active vision module. It consists of a bidirectional XR bridge, a 2-DoF camera mechanism, and asynchronous control and recording processes.

Bidirectional XR Bridge. The bridge streams body, hand, head, and controller signals to the host and returns the robot camera view to the headset.

The XR bridge converts PICO outputs into a common host representation consumed by the motion tracker, hand retargeter, and viewpoint controller. Figure 6 visualizes the reconstructed body skeleton, head pose, and articulated hand keypoints. Keeping these signals in a shared timestamped representation allows the downstream controllers and recorder to associate commands that originate from the same operator motion.

The headset sends the body skeleton, hand keypoints, head pose, and controller inputs to the host through timestamped TCP messages. UDP discovery removes manual address configuration, and WebRTC returns the robot camera stream to the headset. Latest-only queues favor fresh commands over intermediate poses and prevent transient delays from accumulating into a processing backlog. Together with the

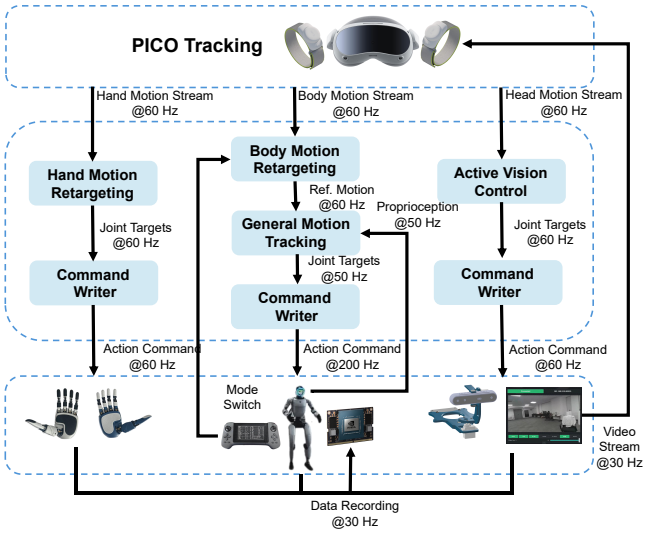

Figure 5: **Deployment architecture.** Asynchronous processes map PICO body, hand, and head streams to robot commands while returning video and recording synchronized data.

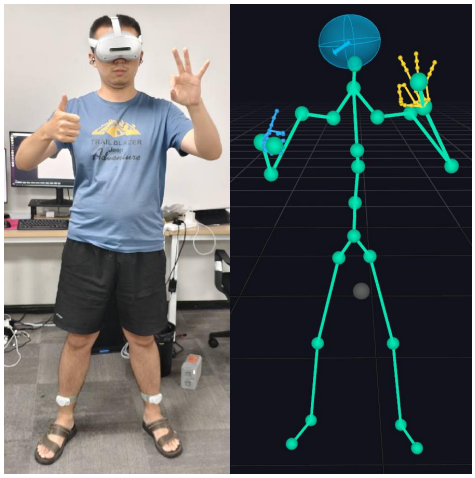

Figure 6: **PICO motion interface.** Host-side visualization of the body skeleton, head pose, and articulated hand keypoints reconstructed from the headset and ankle-tracker signals.

freshness checks described below, the sensing, control, video, and recording processes run asynchronously.

Active Vision Module. The module maps head orientation relative to the torso to yaw and pitch commands after dead-zone filtering and exponential smoothing. Relative head motion prevents torso rotation from changing the commanded viewpoint. Two serial-bus servos actuate the camera, and the returned video closes the viewpoint control loop. The mechanism in Figure 7 costs approximately CNY 500 excluding the camera.

Asynchronous Execution. Separate processes run PICO capture, hand retargeting, viewpoint control, body policy inference, robot communication, video, and recording at their native rates. PICO signals, hand retargeting, and viewpoint control run at 60 Hz; the body policy runs at 50 Hz; low-level body control runs at 200 Hz; and recording runs at 30 Hz. Latest-only communication keeps control responsive, while freshness checks hold the last safe command when a stream becomes stale.

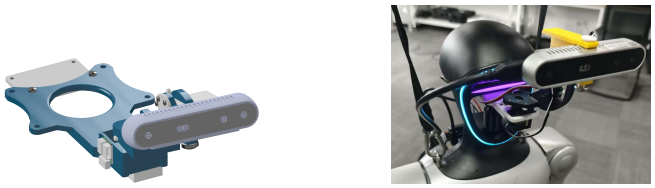

Figure 7: **Active vision module.** CAD and robot-mounted 2-DoF yaw–pitch camera.

4 Experiments

We organize the evaluation around four questions. We first evaluate whether the whole-body motion tracker balances tracking fidelity and robustness, including comparisons, ablations, and deployment on the humanoid. We then examine whether the hand retargeter preserves dexterous poses across different hand morphologies and isolate the effect of its secondary objectives. Next, we evaluate the integrated system through closed-loop viewpoint control and loco-manipulation demonstrations on the humanoid. Finally, we examine whether demonstrations recorded with Teleopit can train policies for autonomous task execution on the humanoid.

4.1 Whole-Body Motion Tracking

The tracker evaluation combines a controlled simulation benchmark with real-world deployment evidence. The benchmark measures tracking accuracy and rollout-level robustness on held-out mocap and live PICO references, while the hardware sequences test whether one policy covers the motions required during teleoperation.

Training Setup

Datasets. We train the tracker on heterogeneous retargeted whole-body motion from three public mocap sources: BONES-SEED [52], TWIST2 [9], and LAFAN1 [53], together with PICO teleoperation recordings captured by our system. Evaluation uses separate mocap and PICO validation subsets to distinguish clean retargeted references from the noisier references encountered during deployment.

The training set combines retargeted public motion with PICO teleoperation recordings to cover diverse motion and the deployment reference distribution. Table 4 reports the source composition. The validation data are disjoint from training and contain a retargeted mocap subset and a recorded PICO subset.

The listed validation sources contain 238 mocap clips and 5 PICO recordings. For evaluation, we retain windows that

Table 4: Training and validation sets for the Teleopit motion tracker. Duration is reported as `h:mm:ss`.

Source	Clips	Duration
Training		
Subset of BONES-SEED	63,237	127:56:57
Subset of TWIST2	31,071	88:14:44
Subset of LAFAN1	57	3:22:58
PICO recordings	9	0:31:19
Validation		
Subset of BONES-SEED	238	0:53:44
PICO recordings	5	0:16:20

span the complete 10-second horizon, which yields 181 mocap windows and 67 PICO windows. This windowing prevents a short source segment from being counted as a successful full-length rollout.

Simulation and training. We train the tracker with PPO in the GPU-accelerated mjlab simulator. The policy combines current-frame features with a temporal encoding of the previous 10 observations, while the critic additionally uses privileged simulation state only during training.

We train the tracker with mjlab on 8 NVIDIA A800 GPUs for approximately 50 hours. The actor and critic use the same architecture: an MLP processes the current observation, a one-dimensional temporal convolution encodes the 10-frame history, and the resulting latent is concatenated with the current-frame features before the policy or value MLP. The critic additionally receives the privileged signals in Table 1; these signals are never used by the deployed actor.

Tracking is anchored at the `torso_link` and covers 14 body links: the pelvis; both hip-roll, knee, and ankle-roll links; the torso; and both shoulder-roll, elbow, and wrist-yaw links. The policy outputs 29 joint-target offsets from the default pose. Table 5 reports the complete simulation, network, and PPO configuration.

Domain randomization and reference sampling. Training combines domain randomization, reference-state initialization, duration-proportional clip sampling, and failure-aware rewind sampling. Section 3.2 and Table 3 give the complete randomization ranges, reference sampling procedure, and termination criteria.

Evaluation Setup

The benchmark contains 181 mocap and 67 PICO windows, each spanning 10 seconds, and every method performs one deterministic rollout per window. We compare against three recent humanoid whole-body trackers: SONIC [38], HoloMotion [39], and TWIST2 [9]. Because their released trackers differ in training data and budgets, we recompute success rate for all methods with the same Teleopit termination conditions. We report success rate over all rollouts and four tracking errors over successful rollouts: MPJPE, root-position error, root-orientation error, and root-velocity error. Results are separated by validation subset in Table 6. The complete evaluation protocol and metric definitions follow.

Table 5: Simulation, network architecture, and PPO hyperparameters of the Teleopit motion tracker.

Setting	Value
Simulation & control	
Simulator	mjlab (GPU MuJoCo)
Robot	Unitree G1, 29 DoF
Physics timestep	5 ms (200 Hz)
Control decimation	4
Policy / PD frequency	50 / 200 Hz
Episode length	10 s
Training hardware	8 NVIDIA A800 GPUs
Parallel environments	8192/GPU (65 536 total)
Training wall time	approximately 50 hours
Architecture	
Actor/critic MLP dims	[2048, 1024, 512, 256, 128]
Activation	ELU
History encoder	Conv1d [256, 128, 64]
History enc. kernel / pool	3 / global average
Observation history length H	10
Observation normalization	empirical (running)
Policy distribution	Gaussian, init. std 1.0
Training (PPO)	
Steps per environment	24
Learning epochs	5
Mini-batches	4
Learning rate	5×10^{-4} (adaptive)
Desired KL	0.01
Discount γ	0.99
GAE λ	0.95
Clip range	0.2
Entropy coefficient	0.005
Value-loss coefficient	1.0
Max gradient norm	1.0
Iterations	40 000

Each comparison method performs one deterministic 10-second rollout on every validation window. We report the mocap and PICO subsets separately. The comparison includes SONIC, HoloMotion, and TWIST2. Their released trackers use different training data, reward functions, training budgets, and termination conditions. At evaluation time, success rate is recomputed for every method using the same Teleopit termination criteria given in the preceding section.

Success rate is computed over every evaluation rollout. Continuous tracking errors are averaged only over rollouts that reach the full 10-second timeout. These errors therefore measure tracking fidelity conditional on completing the full horizon, without mixing early-terminated trajectories of different lengths.

Metric Definitions

Consider a rollout of T control steps that tracks a reference with J=14 links. Let $\mathbf{p}_{t,j} \in \mathbb{R}^3$ denote the position of link j at step t. The vectors $\mathbf{p}_t^{\text{root}}$ and $\mathbf{v}_t^{\text{root}}$ denote the anchor position and body-frame linear velocity, and $\mathbf{q}_t^{\text{root}}$ is its unit orientation quaternion. A hat denotes the corresponding reference quantity. A tilde denotes a quantity expressed in the root heading frame, so global translation and heading are factored out.

Success Rate. Given N evaluation rollouts, success rate (SR) is the fraction that reach the full timeout without early termination:

$$\text{SR} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}[\text{rollout } n \text{ reaches timeout}] \,. \tag{6}$$

Mean Per-Joint Position Error. MPJPE measures pose fidelity after aligning the robot and reference in the root heading frame:

$$\text{MPJPE} = \frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J} \left\|\tilde{\mathbf{p}}_{t,j} - \tilde{\hat{\mathbf{p}}}_{t,j}\right\|_2. \tag{7}$$

Root Position and Orientation Error. These metrics measure global anchor tracking. The orientation error uses the quaternion geodesic angle $\angle(q, \hat{q}) = 2\arccos(|\langle q, \hat{q}\rangle|)$:

$$E_{\text{root}}^{\text{pos}} = \frac{1}{T}\sum_{t=1}^{T} \left\|\mathbf{p}_t^{\text{root}} - \hat{\mathbf{p}}_t^{\text{root}}\right\|_2, \tag{8}$$

$$E_{\text{root}}^{\text{rot}} = \frac{1}{T}\sum_{t=1}^{T} \angle\left(\mathbf{q}_t^{\text{root}}, \hat{\mathbf{q}}_t^{\text{root}}\right). \tag{9}$$

Root Velocity Error. This metric is the framewise body-frame L2 error of anchor linear velocity:

$$E_{\text{root}}^{\text{vel}} = \frac{1}{T}\sum_{t=1}^{T} \left\|\mathbf{v}_t^{\text{root}} - \hat{\mathbf{v}}_t^{\text{root}}\right\|_2. \tag{10}$$

Comparison with Recent Motion Trackers

Results. Teleopit attains the highest SR on both subsets, with 91.7% on mocap and 100.0% on PICO. On the PICO subset, it also obtains the lowest MPJPE, root orientation error, and root velocity error. HoloMotion has the lowest root position error on both subsets, while SONIC has a slightly lower MPJPE on the mocap subset. These results indicate that Teleopit provides a favorable balance between tracking fidelity and stability on PICO references.

Ablation Studies

We ablate the two components that distinguish the tracker from a standard PPO controller: (i) *w/o history encoder*, which removes the temporal Conv1d encoder and feeds only the current-frame observation; and (ii) *w/o rewind sampling*, which replaces failure-aware rewind with uniform clip resampling (rewind_prob = 0).

All three variants use a shared reduced training setting: four GPUs, 4096 environments per GPU (16 384 total), and 20k iterations. The data, PPO hyperparameters, and evaluation protocol are held fixed across the variants, which are evaluated on the mocap validation subset. Table 7 reports their within-setting differences; Table 6 separately reports the final 8-GPU/40k checkpoint.

Results. The full configuration attains the highest success rate (74.0%). Removing the history encoder reduces both tracking fidelity and success. Removing rewind sampling reduces success and root-orientation accuracy, although some position metrics improve slightly. Rewind repeatedly revisits failure-adjacent segments, which also helps the policy learn difficult motions earlier in training.

Real-Robot Tracking

Whole-body tracking on hardware. Figure 8 pairs the operator and robot across static poses, running, sidestepping, turning, and large changes in base height. The same policy tracks all motions without motion-specific switching. The kneel–stand and sit–stand sequences further demonstrate continuous transitions through intermediate configurations on the humanoid.

4.2 Dexterous Hand Retargeting

The hand evaluation tests two complementary claims: the shared objective transfers human poses across different robot-hand morphologies, and its distance and frame objectives target complementary fingertip-distance and thumb-frame alignment errors.

Quantitative Protocol and Metrics

Evaluation protocol. All six robot hands are evaluated on the same right-hand PICO recording. The recording lasts 25.8 seconds and was captured at 80 Hz. In the metrics below, T denotes the number of time samples. All hands share the objective weights, activation thresholds, preprocessing parameters, filter, and SLSQP settings. The distance target uses raw human distances with scale 1.0.

Ablation protocol. Objective ablations use the LinkerHand L20 right hand and hold the input, initialization, joint limits, remaining objectives, filter, and solver settings fixed.

Timing protocol. We run an untimed warm-up pass before measurement. Solve time covers the SLSQP call and excludes sensing, visualization, communication, and robot control.

Table 6: Motion-tracking comparison on the validation set, reported separately for clean retargeted mocap references and the noisier PICO teleoperation references. Best per column in **bold**.

Method	Mocap validation					PICO validation				
	SR ↑ (%)	MPJPE ↓ (m)	R. pos. ↓ (m)	R. ori. ↓ (rad)	R. vel. ↓ (m/s)	SR ↑ (%)	MPJPE ↓ (m)	R. pos. ↓ (m)	R. ori. ↓ (rad)	R. vel. ↓ (m/s)
TWIST2 [9]	43.1	0.248	0.372	0.235	0.245	64.2	0.209	1.003	0.292	0.417
SONIC [38]	75.7	0.037	0.286	**0.108**	0.224	82.1	0.038	0.575	0.104	0.272
HoloMotion [39]	64.6	**0.034**	**0.103**	0.227	**0.205**	97.0	0.035	**0.127**	0.244	0.273
Teleopit	**91.7**	0.040	0.260	0.109	0.228	**100.0**	**0.029**	0.523	**0.089**	**0.262**

Figure 8: **Real-robot whole-body tracking.** Static motions and continuous kneel–stand and sit–stand transitions on hardware.

Table 7: Tracker ablations on the mocap validation subset. All rows use a shared reduced setting of four GPUs, 4096 environments/GPU, and 20k iterations. Table 6 reports the final checkpoint. Best per column in **bold**.

Variant	SR ↑ (%)	MPJPE ↓ (m)	R. pos. ↓ (m)	R. ori. ↓ (rad)	R. vel. ↓ (m/s)
Full (reduced)	**74.0**	0.045	0.356	**0.118**	0.266
w/o rewind	72.9	**0.042**	**0.305**	0.124	**0.235**
w/o history	73.5	0.046	0.356	0.124	0.263

Continuous pose metrics. Direction error averages the angular error over the five semantic finger-base-to-tip rays shared by every hand:

$$E_{\text{dir}} = \frac{1}{5T}\sum_{t=1}^{T}\sum_{i=1}^{5}\arccos\Big(\text{clip}\left(\hat{\mathbf{v}}_{t,i}^{R}\cdot\hat{\mathbf{v}}_{t,i}^{H},-1,1\right)\Big). \tag{11}$$

Here, $\hat{\mathbf{v}}_{t,i}^{H}$ and $\hat{\mathbf{v}}_{t,i}^{R}$ are unit human and robot rays. Using a common set of semantic rays prevents hands with different internal objective topologies from changing the evaluation dimension. We report E_{dir} in degrees.

Normalized semantic keypoint error compares the five fingertips after translation by the wrist or palm base and normalization by the corresponding wrist-to-middle-base or palm-base-to-middle-base scale:

$$E_{\text{key}} = \frac{1}{5T}\sum_{t=1}^{T}\sum_{k=1}^{5}\left\|\frac{\mathbf{p}_{t,k}^{R}-\mathbf{p}_{t,w}^{R}}{s_R}-\frac{\mathbf{p}_{t,k}^{H}-\mathbf{p}_{t,w}^{H}}{s_H}\right\|_2. \tag{12}$$

Site-distance tracking error averages the absolute error between human and robot thumb-to-fingertip distances for the other four fingertips $\mathcal{P}=\{\text{index},\text{middle},\text{ring},\text{little}\}$:

$$d_{t,p}^{X} = \left\|\mathbf{p}_{t,\text{thumb}}^{X}-\mathbf{p}_{t,p}^{X}\right\|_2, \quad X\in\{H,R\}, \tag{13}$$

$$E_{\text{dist}} = \frac{1}{4T}\sum_{t=1}^{T}\sum_{p\in\mathcal{P}}\left|d_{t,p}^{R}-d_{t,p}^{H}\right|. \tag{14}$$

Thumb-frame error is the geodesic distance between human and robot rotation matrices constructed from the thumb-base primary and secondary axes:

$$\begin{aligned} c_t &= \frac{\text{tr}\left((\mathbf{R}_t^R)^\top\mathbf{R}_t^H\right)-1}{2},\\ e_{\text{frame},t} &= \arccos(\text{clip}(c_t,-1,1)),\\ E_{\text{frame}} &= \frac{1}{T}\sum_{t=1}^{T}e_{\text{frame},t}. \end{aligned} \tag{15}$$

We report E_{frame} in degrees. Mean solve time is the average duration of the SLSQP call over the recording.

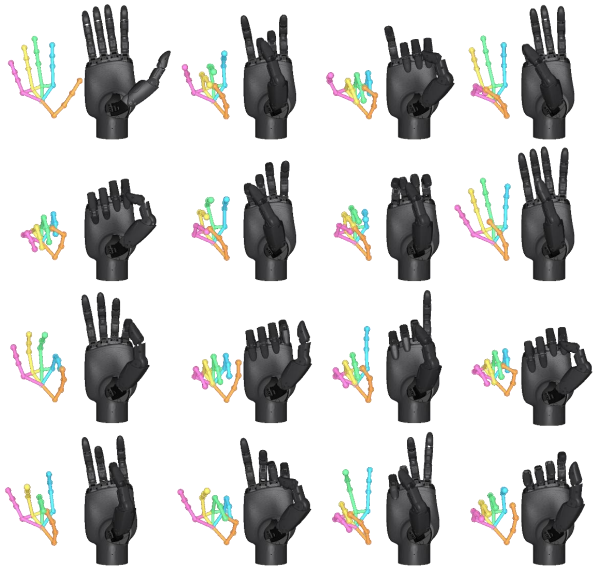

Figure 9: **Pose fidelity across gestures.** Human skeletons and retargeted poses show finger-direction, pinch, and thumb-opposition fidelity.

Table 8: Cross-hand evaluation with shared objective and solver parameters. Active joints are independently optimized; passive joints follow active joints through fixed mimic couplings. All error and time metrics are lower-is-better.

Hand	Active	Passive	E_{dir} (°)	E_{key}	E_{dist} (mm)	Mean solve (ms)
Dex5	20	0	9.08	1.42	9.46	3.13
Inspire DFQ	6	6	29.53	1.98	14.45	1.56
LinkerHand L20	16	5	14.67	1.83	19.82	4.27
LinkerHand L6	6	5	19.43	1.74	21.15	1.93
Rohand	6	19	43.04	1.63	14.57	8.53
Sharpa Wave	22	0	9.02	1.40	18.05	4.04

Pose Fidelity and Cross-Embodiment Transfer

Pose fidelity. Figure 9 retargets a range of operator hand poses onto a single dexterous hand. Across free-hand gestures, pinches, and opposed-thumb grasps, the solved configuration preserves the finger directions and follows fingertip closure without per-pose tuning.

Cross-embodiment transfer. Figure 10 applies the same objective form to a library of commercial dexterous hands. Their proportions and thumb layouts differ substantially. The normalized-direction formulation transfers across these hands without per-vector human-to-robot scale tuning.

Quantitative cross-hand results. Table 8 evaluates the shared numerical settings on the full recording for each hand. The same objective and solver parameters run across all six morphologies despite their different active and passive joint counts. Mean solve time ranges from 1.56 to 8.53 ms.

Cross-hand accuracy is not uniform. Direction error ranges from 9.02° to 43.04°, while mean site-distance tracking error ranges from 9.46 to 21.15 mm. The experiment therefore supports shared parameterization across the six tested morphologies, while showing that retargeting accuracy remains morphology dependent.

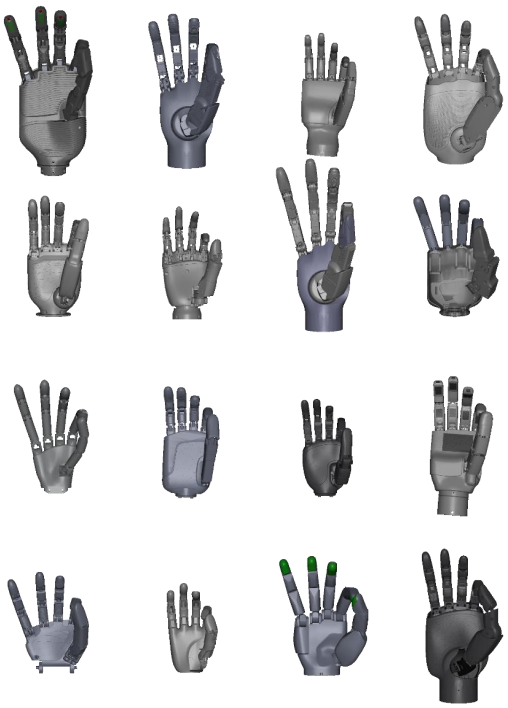

Figure 10: **Transfer across dexterous hands.** One objective handles different link lengths, finger counts, and thumb layouts; only semantic links and mechanical couplings change.

Table 9: Distance-objective ablation on LinkerHand L20. All metrics are lower-is-better; best per column in **bold**.

Variant	E_{dist} (mm)	E_{dir} (°)	E_{key}	Mean solve (ms)
w/o distance	48.46	**11.89**	**1.80**	**2.93**
Full	**19.82**	14.67	1.83	4.27

Table 10: Thumb-frame-objective ablation on LinkerHand L20. All metrics are lower-is-better; best per column in **bold**.

Variant	E_{frame} (°)	E_{dir} (°)	E_{key}	Mean solve (ms)
w/o frame	31.53	15.05	1.86	**3.45**
Full	**22.33**	**14.67**	**1.83**	4.27

Objective Ablations

Figure 11 isolates the two secondary objectives on representative poses. Removing the distance objective weakens fingertip closure, while removing the thumb-base frame objective weakens opposition roll. In a shared-parameter evaluation, the two objectives reduce their respective distance and frame errors by 59.1% and 29.2%.

Figure 11: **Retargeting ablation.** The distance objective maintains pinch closure, and the frame objective preserves thumb opposition.

Distance objective. Table 9 shows that the distance objective reduces mean site-distance tracking error from 48.46 to 19.82 mm, a 59.1% reduction. This improvement trades off against a 2.78° increase in direction error, a 0.03 increase in normalized keypoint error, and a 1.34 ms increase in mean solve time. The result shows that the distance objective improves continuous fingertip-distance tracking under the fixed ablation protocol.

Thumb-frame objective. Table 10 shows that the frame objective reduces mean thumb-frame error from 31.53° to 22.33°, a 9.20° or 29.2% reduction. Direction and normalized keypoint errors also decrease by 0.38° and 0.03, respectively, while mean solve time increases by 0.82 ms.

4.3 Integrated Real-Robot Teleoperation

We finally verify that the evaluated components operate together on the humanoid during coordinated loco-manipulation.

End-to-End Latency

We estimate latency from paired arrival events visible in the same video recording. All network communication during the four measurements uses Wi-Fi. For each path, latency is the elapsed time from the initiating motion to the first visible downstream response. This procedure measures the complete path to the output shown in Figure 12, including the sensing, communication, control, and rendering contributions along that path.

The four measured paths respond within approximately 0.05–0.15 s under the recorded conditions. Timing uncertainty is bounded by the video sampling interval and manual labeling of the first visible response. For dexterous retargeting, latency ends when the retargeted hand configuration first appears on the display.

Closed-Loop Viewpoint Control

The head-driven module provides live yaw–pitch viewpoint adjustment while the operator continues to control the body and hands.

Figure 7 shows the yaw–pitch hardware and summarizes its closed-loop function. Figure 13 provides the corresponding camera observations across the tested workspace. The neutral view appears at the center, while the surrounding images cover vertical, horizontal, and diagonal head commands. The diagonal views show simultaneous yaw and pitch motion.

Integrated Loco-Manipulation

Figure 14 exercises the complete sensing and control path rather than any component in isolation. The demonstrations span tabletop organization, transferring objects with a bag, operating lights and doors, and sorting shelf items. They require the operator to coordinate base placement and posture with arm reach, hand closure, and viewpoint selection, showing that the asynchronous multi-rate runtime preserves the coupling needed for extended whole-body tasks.

4.4 Policy Learning from Teleopit Demonstrations

We examine whether demonstrations recorded with Teleopit can directly support autonomous whole body manipulation. We train ACT [54] and GR00T N1.7 [7] on the same recordings. During autonomous execution, the learned policy replaces the human operator at the reference motion interface, while the motion tracker and robot control stack remain unchanged. This design tests both the recorded data and the interface through which task policies control the complete robot.

Action Space and Control Interface

Design motivation. A task policy must coordinate the floating base, body joints, dexterous hands, and active viewpoint. Predicting low level motor targets would additionally require the task policy to learn balance and dynamic tracking from only a small task dataset. Teleopit already solves this problem by converting a kinematic whole body reference into feasible robot motion. We therefore train the task policy to predict the same reference quantities produced during teleoperation. The motion tracker remains responsible for whole body tracking, whereas the hand and neck joint references are handled by their respective controllers. The action space consequently preserves the control hierarchy used to record the demonstrations.

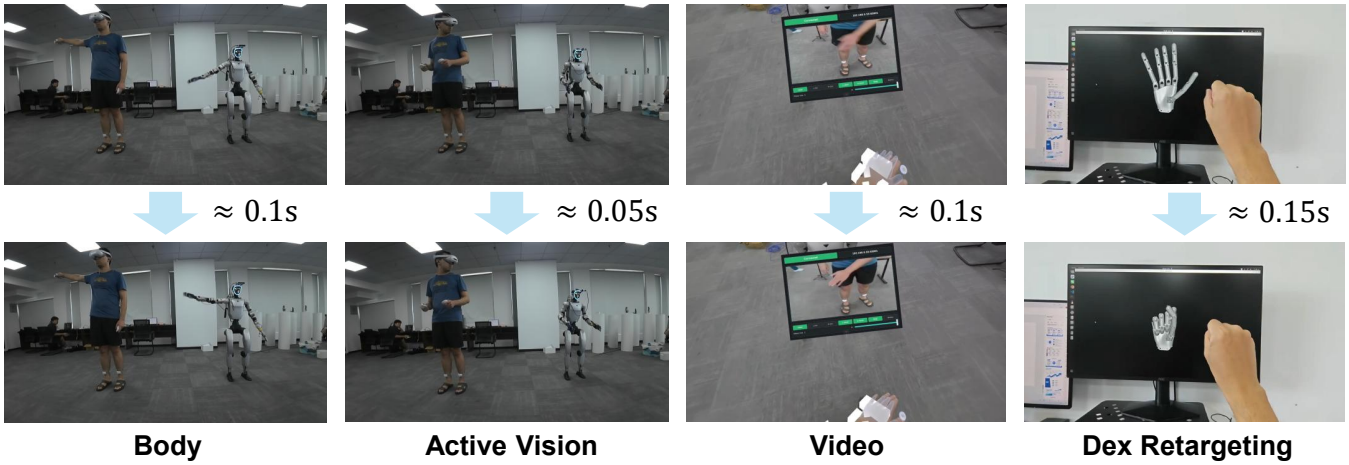

Figure 12: **Video-based estimates of end-to-end latency.** Paired images show the initiating event (top) and the downstream arrival (bottom). The estimated delays are approximately 0.10 s for whole-body response, 0.05 s for viewpoint-control response, 0.10 s for the video stream arriving at the PICO headset, and 0.15 s from human-hand motion to the displayed retargeted dexterous-hand configuration.

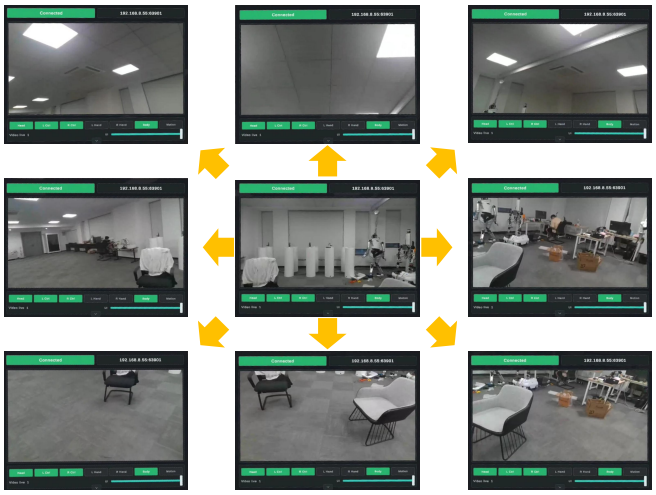

Figure 13: **Head-driven viewpoint control.** Views returned to the headset as the operator looks around from the neutral view (center). The eight surrounding frames show the camera coverage obtained by commanding yaw and pitch through head motion.

Observation and recorded reference. At step t, the policy receives an RGB image $\mathbf{I}_t$ and the measured robot state

$$\mathbf{s}_t = [\mathbf{q}_t^{\text{body}}, \mathbf{q}_t^{\text{hand}}, \mathbf{q}_t^{\text{neck}}] \in \mathbb{R}^{43}, \tag{16}$$

which contains 29 body joint positions, 12 dexterous hand joint positions, and two neck joint positions. Each recorded action is a 50D reference

$$\mathbf{a}_t^{\text{ref}} = [\mathbf{p}_t^{\text{root}}, \mathbf{r}_t^{\text{root}}, \bar{\mathbf{q}}_t^{\text{body}}, \bar{\mathbf{q}}_t^{\text{hand}}, \bar{\mathbf{q}}_t^{\text{neck}}] \in \mathbb{R}^{50}. \tag{17}$$

The first two terms specify the 3D root position and 4D root orientation. The remaining terms specify 29 body, 12 hand, and two neck joint references. The root pose and body joints define the reference trajectory for the motion tracker; the hand and neck terms preserve the commands recorded for the other two robot subsystems.

Policy action. Global planar position and heading depend on where the robot starts and do not describe the task itself. Learning them as absolute values would introduce irrelevant variation across demonstrations. We instead express root motion relative to the robot pose at the beginning of each predicted chunk. For a future step $t + k$,

$$\Delta\mathbf{p}_{t,k}^{xy} = \mathbf{R}_z(-\psi_t)(\mathbf{p}_{t+k}^{xy} - \mathbf{p}_t^{xy}), \tag{18}$$

$$\Delta\mathbf{R}_{t,k} = \mathbf{R}_t^\top \mathbf{R}_{t+k}. \tag{19}$$

The policy action contains the 2D planar displacement $\Delta\mathbf{p}_{t,k}^{xy}$, absolute root height p_{t+k}^z, and the continuous 6D representation of $\Delta\mathbf{R}_{t,k}$ from Zhou et al. [55], followed by the 43 body, hand, and neck joint references. This gives a 52D policy action. Root height remains absolute because it is defined with respect to the ground plane. Using one source pose for the complete chunk also preserves a consistent relative trajectory across its predicted steps.

Figure 15 summarizes the deployment interface. The root pose associated with the current observation converts the relative root action back to the robot frame. The reconstructed body trajectory is supplied to the 50 Hz motion tracker, which produces joint targets tracked by the 200 Hz PD controller. Hand and neck references bypass the motion tracker and are sent directly to their corresponding controllers.

Training and Inference Setup

We convert the 96 successful 30 Hz recordings into a LeRobot dataset [56] and train the two policies independently. Both policies use the observation and action spaces defined above. Table 11 reports the settings that affect optimization

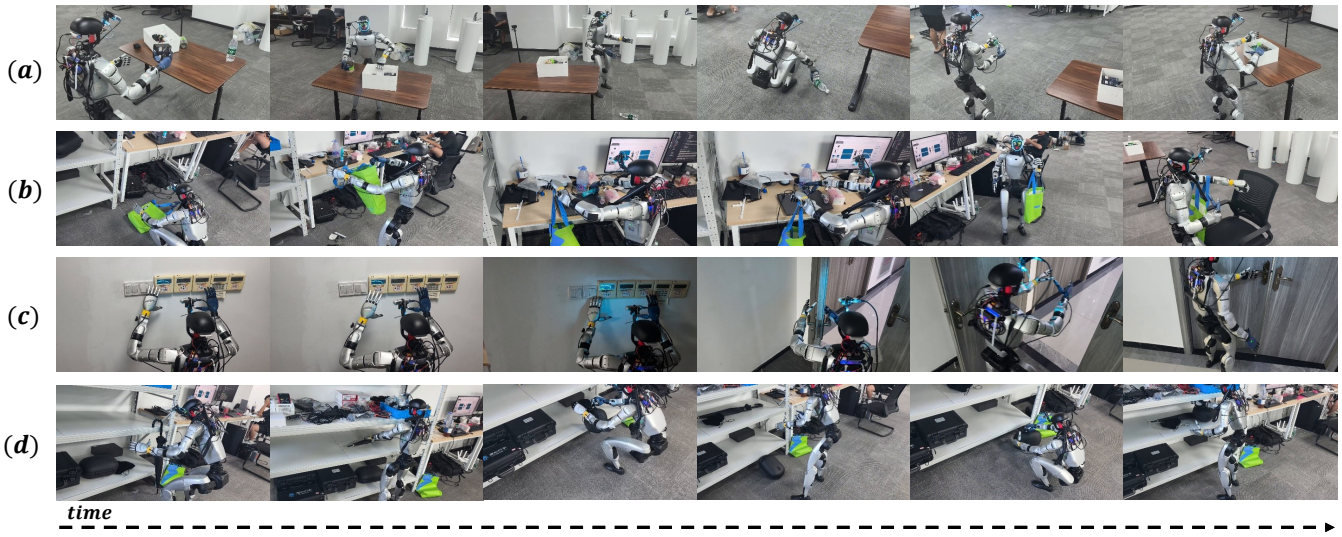

Figure 14: **Integrated loco-manipulation.** Body, hand, and vision control across four household tasks.

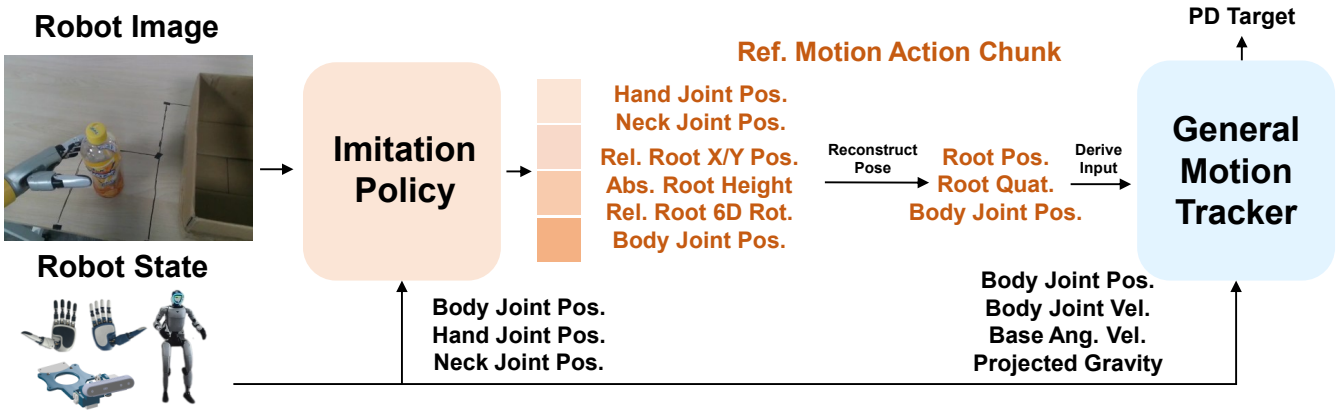

Figure 15: **Hierarchical interface for autonomous execution.** The task policy predicts a 52D reference motion chunk from the robot image and measured joint state. The initial root pose converts the relative root prediction back to a 50D robot reference. Root and body references enter the Teleopit motion tracker, which produces joint targets for the 200 Hz PD controller. Dexterous hand and neck joint references are sent to their corresponding controllers.

Table 11: Training settings used for ACT and GR00T N1.7. Batch size is reported per device.

Setting	ACT	GR00T N1.7
Initialization	ImageNet R18	N1.7 3B
Optimizer updates	150,000	20,000
Micro batch	32	1
Accumulation steps	1	8
Action chunk length	50	40
Learning rate	1×10^{-5}	1×10^{-4}
Weight decay	1×10^{-4}	1×10^{-5}
Numeric precision	FP32	BF16

Table 12: Inference and control settings shared by both policies.

Setting	Value
Visual observation	640×480 RGB at 30 Hz
Policy action dimension	52
Executed steps before replanning	20
Motion tracker rate	50 Hz
Low level PD rate	200 Hz

or model capacity; data loading and checkpointing parameters are omitted.

ACT uses its ImageNet initialized ResNet18 visual encoder and is optimized in full precision. For GR00T N1.7, the language and visual backbones remain frozen, while the projector and diffusion model are optimized with BF16. These choices preserve the pretrained representations of the foundation model while adapting its action generation to the Teleopit embodiment and reference space.

As summarized in Table 12, we use asynchronous chunk inference native to both policies. After executing 20 predicted steps, the system acquires a new observation and replans. Policy inference runs concurrently with robot control, so the

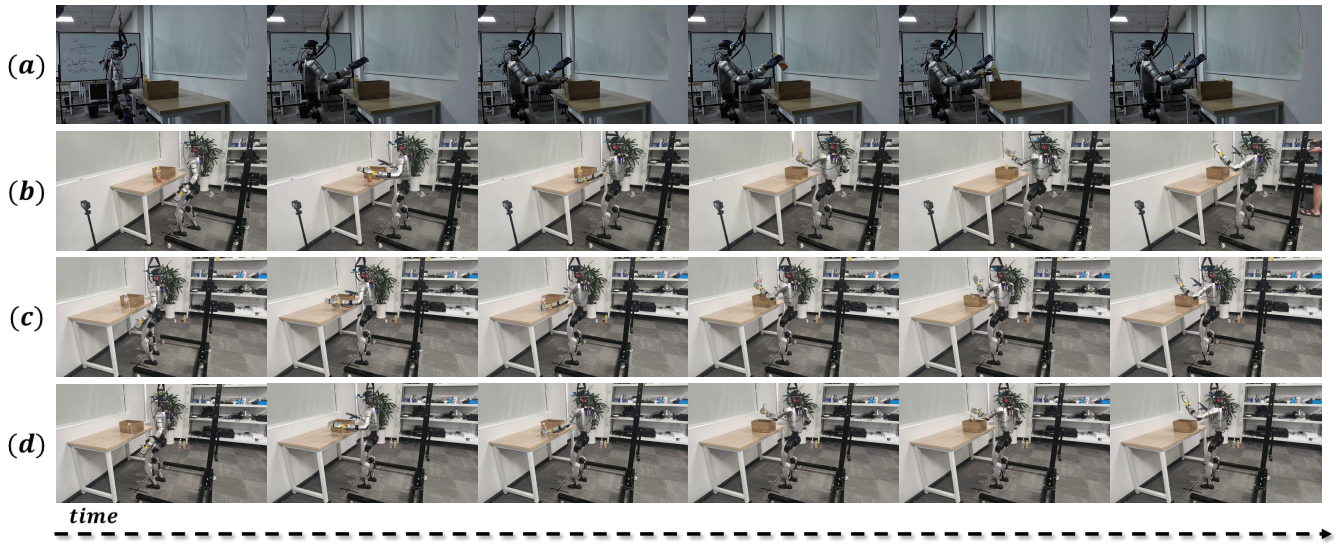

Figure 16: **Teleoperation collection and autonomous deployment.** Each row proceeds from left to right. (a) Teleopit records an operator demonstration of grasping a bottle and placing it inside the adjacent box. (b) GR00T N1.7 autonomously executes the same task. (c,d) Without further demonstrations or fine tuning, the policy completes the task with two bottle types absent from its training demonstrations.

Table 13: Teleoperation collection yield and task success rates on the bottle-placement task. Both learned policies use the same 96 demonstrations.

Control	Training set size	Success / trials	SR (%)
Teleoperation	–	96 / 100	96.0
ACT	96	18 / 20	90.0
GR00T N1.7	96	19 / 20	95.0

current reference continues to drive the motion tracker until the next chunk is available. ACT and GR00T therefore differ in model architecture and prediction length but share the same execution schedule and low level control stack.

Data Collection and Autonomous Deployment

Task and protocol. The task requires the robot to grasp a bottle from a table and place it inside an adjacent box. We record 100 teleoperated trials at 30 Hz. A collection trial is successful when the operator completes the task, and only successful trials are included in the training set. We then evaluate each trained policy in 20 trials on the humanoid under the same task setup. A trial is counted as successful if the bottle remains inside the box at the end of the rollout without operator intervention.

Quantitative results. Table 13 reports that teleoperation succeeds in 96 of 100 collection trials, yielding 96 demonstrations for policy learning. Trained on this shared dataset, ACT succeeds in 18 of 20 autonomous trials and GR00T N1.7 succeeds in 19 of 20. Thus, the same recordings and reference interface support both a task specific action chunking policy and a vision language action model at success rates of at least 90%.

Qualitative results. Figure 16(a) presents the teleoperated demonstration, while Figure 16(b) shows that the learned policy reproduces its coordinated whole body reach, grasp, transport, and placement. Figure 16(c,d) further shows successful execution with bottles of different appearance and geometry, without additional training. These examples qualitatively demonstrate zero-shot transfer across bottle instances for the same task.

5 Conclusion

We presented Teleopit, a full-embodiment humanoid teleoperation system that uses VR to command whole-body motion, continuous dexterous hand motion, and viewpoint control. The system integrates a history-aware motion tracker, an optimization-based hand retargeter, an active vision module, and an asynchronous runtime. Normalized finger directions allow the hand objective to transfer across morphologies without hand-specific hyperparameter tuning, while distance and thumb-frame objectives reduce complementary closure and alignment errors.

The tracker achieves success rates of 91.7% and 100.0% on the mocap and PICO validation subsets, respectively. Retargeting experiments cover more than a dozen dexterous hands, and real-robot teleoperation combines locomotion, continuous hand control, and viewpoint control. Demonstrations recorded with Teleopit also support downstream policy learning. Trained on a shared set of 96 demonstrations, ACT and GR00T N1.7 achieve task success rates of 90.0% and 95.0%, respectively, when deployed on the humanoid.

References

[1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[2] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[3] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[4] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[5] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[7] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[8] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[9] Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C Karen Liu. Twist2: Scalable, portable, and holistic humanoid data collection system. *arXiv preprint arXiv:2511.02832*, 2025.

[10] Liang Heng, Yihe Tang, Jiajun Xu, Henghui Bao, Di Huang, and Yue Wang. Humdex: Humanoid dexterous manipulation made easy. *arXiv preprint arXiv:2603.12260*, 2026.

[11] Zhigen Zhao, Liuchuan Yu, Ke Jing, and Ning Yang. Xrobotoolkit: A cross-platform framework for robot teleoperation. In *2026 IEEE/SICE International Symposium on System Integration (SII)*, pages 15–20. IEEE, 2026.

[12] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[13] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.015. URL https://doi.org/10.15607/RSS.2023.XIX.015.

[14] Zhao-Heng Yin, Changhao Wang, Luis Pineda, Krishna Bodduluri, Tingfan Wu, Pieter Abbeel, and Mustafa Mukadam. Geometric retargeting: A principled, ultrafast neural hand retargeting algorithm. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 17376–17382. IEEE, 2025.

[15] Kevin Zakka, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for gpu-accelerated robot learning. *arXiv preprint arXiv:2601.22074*, 2026.

[16] Nathan Miller, Odest Chadwicke Jenkins, Marcelo Kallmann, and Maja J. Mataric. Motion capture from inertial sensing for untethered humanoid teleoperation. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2004.

[17] Francisco-Javier Montecillo-Puente, Manish N. Sreenivasa, and Jean-Paul Laumond. On real-time whole-body human to humanoid motion transfer. In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pages 22–31, 2010. doi: 10.5220/0002915300220031. URL https://doi.org/10.5220/0002915300220031.

[18] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024. doi: 10.1109/IROS58592.2024.10801984. URL https://doi.org/10.1109/IROS58592.2024.10801984.

[19] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 1516–1540. PMLR, 2025. URL https://proceedings.mlr.press/v270/he25b.html.

[20] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2828–2844. PMLR, 2025. URL https://proceedings.mlr.press/v270/fu25a.html.

[21] Yanjie Ze, Zixuan Chen, JoÃĢo Pedro AraÃšjo, Ziang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025.

[22] Mingyo Seo, Steve Han, Kyutae Sim, Seung Hyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8, 2023. URL https://arxiv.org/abs/2309.01952.

[23] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

[24] Yixuan Li, Yutang Lin, Jieming Cui, Tengyu Liu, Wei Liang, Yixin Zhu, and Siyuan Huang. Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks. In *9th Annual Conference on Robot Learning*, 2025.

[25] Ruiqian Nai, Boyuan Zheng, Junming Zhao, Haodong Zhu, Sicong Dai, Zunhao Chen, Yihang Hu, Yingdong Hu, Tong Zhang, Chuan Wen, et al. Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations. *arXiv preprint arXiv:2602.06643*, 2026.

[26] Chenhao Yu, Hongwu Wang, Youhao Hu, Jiachen Zhang, Yuanyuan Li, and Shaqi Luo. Bifrostumi: Bridging robot-free demonstrations and humanoid whole-body manipulation. *arXiv preprint arXiv:2605.03452*, 2026.

[27] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4): 1–14, 2018. doi: 10.1145/3197517.3201311. URL https://doi.org/10.1145/3197517.3201311.

[28] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40(4):1–20, 2021. doi: 10.1145/3450626.3459670. URL https://doi.org/10.1145/3450626.3459670.

[29] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.

[30] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *International Conference on Learning Representations*, volume 2024, pages 56766–56782, 2024.

[31] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[32] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[33] Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9989–9996. IEEE, 2025.

[34] Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv preprint arXiv:2506.14770*, 2025.

[35] Zhikai Zhang, Jun Guo, Chao Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Han Xue, Zhenrong Wang, Maoqi Liu, Jiangran Lyu, et al. Track any motions under any disturbances. *arXiv preprint arXiv:2509.13833*, 2025.

[36] Yubiao Ma, Han Yu, Jiayin Xie, Changtai Lv, Qiang Luo, Chi Zhang, Yunpeng Yin, Boyang Xing, Xuemei Ren, and Dongdong Zheng. Robust and generalized humanoid motion tracking. *arXiv preprint arXiv:2601.23080*, 2026.

[37] Jie Li, Bing Tang, and Feng Wu. Telegate: Whole-body humanoid teleoperation via gated expert selection with motion prior. *arXiv preprint arXiv:2602.09628*, 2026.

[38] Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

[39] Maiyue Chen, Kaihui Wang, Bo Zhang, Xihan Ma, Zhiyuan Yang, Yi Ren, Qijun Huang, Zihao Zhu, Yucheng Wang, and Zhizhong Su. Holomotion-1 technical report. *arXiv preprint arXiv:2605.15336*, 2026.

[40] Haiying Hu, Xiaohui Gao, Jiawei Li, Jie Wang, and Hong Liu. Calibrating human hand for teleoperating the hit/dlr hand. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 5, pages 4571–4576, 2004. doi: 10.1109/ROBOT.2004.1302438. URL https://doi.org/10.1109/ROBOT.2004.1302438.

[41] Cassie Meeker, Thomas Rasmussen, and Matei Ciocarlie. Intuitive hand teleoperation by novice operators using a continuous teleoperation subspace. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5821–5827, 2018. doi: 10.1109/ICRA.2018.8460506. URL https://doi.org/10.1109/ICRA.2018.8460506.

[42] Roberto Meattini, Dario Chiaravalli, Luigi Biagiotti, Gianluca Palli, and Claudio Melchiorri. Combined joint-cartesian mapping for simultaneous shape and precision teleoperation of anthropomorphic robotic hands. *IFAC-PapersOnLine*, 53(2):10052–10057, 2020. doi: 10.1016/j.ifacol.2020.12.2726. URL https://doi.org/10.1016/j.ifacol.2020.12.2726.

[43] Roberto Meattini, Raúl Suárez, Gianluca Palli, and Claudio Melchiorri. Human to robot hand motion mapping methods: Review and classification. *IEEE Transactions on Robotics*, 39(2):842–861, 2023. doi: 10.1109/TRO.2022.3205510. URL https://doi.org/10.1109/TRO.2022.3205510.

[44] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. In *Proceedings of Robotics: Science and Systems*, 2022. URL https://arxiv.org/abs/2202.10448.

[45] Chenxi Wang, Ying Feng, Hongjie Fang, Shangning Xia, Lixin Yang, Chuan Wen, and Cewu Lu. Anydexrt: Calibration-free dexterous hand retargeting with few-shot human guidance. *arXiv preprint arXiv:2607.08341*, 2026.

[46] Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025.

[47] Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. Spider: Scalable physics-informed dexterous retargeting. *arXiv preprint arXiv:2511.09484*, 2025.

[48] Dongmyoung Lee, Chengxi Li, and Dongheui Lee. Dextwist: Dexterous hand retargeting for twist motion via mixed reality-based teleoperation. In *2026 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*, pages 149–154. IEEE, 2026.

[49] Liyuan Qi, Olaoluwa Popoola, Muhammad Ali Imran, and Wasim Ahmad. Genhand: Generalised human grasp kinematic retargeting. *npj Robotics*, 4, 2026. doi: 10.1038/s44182-026-00076-1. URL https://doi.org/10.1038/s44182-026-00076-1.

[50] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[51] Ruoshi Wen, Jiajun Zhang, Guangzeng Chen, Zhongren Cui, Min Du, Yang Gou, Zhigang Han, Junkai Hu, Liqun Huang, Hao Niu, et al. Dexterous teleoperation of 20-dof bytedexter hand via human motion retargeting. *arXiv preprint arXiv:2507.03227*, 2025.

[52] BONES Studio. BONES-SEED: Skeletal everyday embodiment dataset. https://bones.studio/datasets/seed, 2026. Accessed 2026-07-14.

[53] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics*, 39(4), 2020. doi: 10.1145/3386569.3392480. URL https://arxiv.org/abs/2102.04942.

[54] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

[55] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[56] Rémi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Mustafa Shukor, Jess Moss, Alexander Soare, Dana Aubakirova, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. LeRobot: An open-source library for end-to-end robot learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.22818.